

# Energy-Efficient Digital Design through Inexact and Approximate Arithmetic Circuits

Vincent Camus<sup>†</sup>, Jeremy Schlachter<sup>†</sup>, Christian Enz

Integrated Circuits Laboratory (ICLAB)

Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

vincent.camus@epfl.ch, jeremy.schlachter@epfl.ch

<sup>†</sup>these authors contributed equally to this work

**Abstract**—Inexact and approximate circuit design is a promising approach to improve performance and energy efficiency in technology-scaled and low-power digital systems. Such strategy is suitable for error tolerant applications involving perceptive or statistical outputs. This paper reviews two established techniques applicable to arithmetic units: circuit pruning and carry speculation. A critical comparative study is carried out considering several error metrics.

## I. INTRODUCTION

Approximate computing has become a major field of research in the sense that it could significantly improve energy efficiency and performances of modern digital circuit. It is also a potential solution to overcome the limitations of technology scaling and the forecasted end of Moore's law. To that extent, many inexact or approximate circuits have been presented in the literature, and most of them are based on manual design or tweaks, hardly integrable in a standard digital flow.

This article aims at reviewing two techniques that can easily be implemented in a standard digital flow: probabilistic pruning and carry speculation. The remainder of this paper is organised as follow: section II reminds the state-of-the-art for the two design techniques, section III re-defines the error metrics and provides a functional description of the pruning and the speculation techniques, and finally section IV carries out a comparative study considering several error metrics.

## II. STATE-OF-THE-ART

### A. Probabilistic Pruning

Probabilistic pruning is a design technique consisting in removing circuit blocks or elements such as full adder cells, gate clusters or single gates in order to trade exactness of computation against power, area and delay savings without any overhead. The decision of pruning one of these elements is based on two parameters: the significance, which is a structural parameter, and the activity determined by hardware simulations. The amount of error is simply proportional to the number of pruned elements. This technique was first introduced in [1], where full adder cells were removed from various adder architectures, resulting in gains of up to 7.5 X in Energy-Delay-Area Product (EDAP), for a 10 % relative error magnitude.

Later on, this technique has been improved by integrating it in the standard digital flow using existing tools, and by applying

pruning at the gate level [2]. This finer granularity enables an order of magnitude area and power savings for a 64-bit adder with 10 % relative error magnitude. It has also been shown that 25 % power and area reduction can be achieved for 16-bit multipliers.

### B. Speculative Adders

Speculative adders [3] exploit the fact that carry propagate sequences in additions are typically short, making it possible to estimate intermediate carries using a limited number of previous stages. They split the binary addition into several subpaths executed concurrently for higher execution speed and energy efficiency, but at the risk of generating occasionally incorrect results. Thus, the critical path of the adder can be divided in two or more shorter paths, relaxing constraints over the entire design and improving the speed, area and power beyond the theoretical bounds of exact adders.

A number of speculative adders have been proposed in literature with different approaches in order to reduce the error frequency or magnitude. The ETAII adder [4] consists of regular sub-adder blocks with input carries speculated from Carry Look Ahead (CLA) blocks of the same length. In the ETAIIM version, several of the most significant CLA blocks are chained to increase accuracy. The ETBA adder [5], direct descendant of the ETAIIM, adds variable speculation signs and sub-adder sum balancing multiplexed blocks to mitigate relative errors. The ETAIV [6] and CSA [7] adders have enhanced accuracy by considering two prior carry speculation blocks instead of one, coupled respectively with a carry select or a carry skip technique, with the latter also using sum balancing over several sub-adder blocks. On the other side, ISA [8] and CSC [9] adders have recently improved circuit performance and efficiency by introducing off-critical path error reduction techniques. The ISA adder concept [8] has also proposed an optimal and generalized approach of speculative compensated adders, encompassing aforementioned adders, and has introduced a simple methodology to allow designers to generate efficient architectures from a delay-accuracy specification.

## III. INEXACT ARITHMETIC CIRCUIT DESIGN

### A. Gate-Level Pruning

Gate-Level Pruning is a CAD technique to automatically generate inexact circuits starting from a conventional design by adding only one small step in the digital design flow. The CAD framework is presented in Fig. 1.

Any exact circuit can be represented by a directed acyclic graph as depicted in Fig. 2, where the nodes are components

This work was supported by the NanoTera "IcySoC" project and the Swiss National Science Foundation grant No 200021-144418.

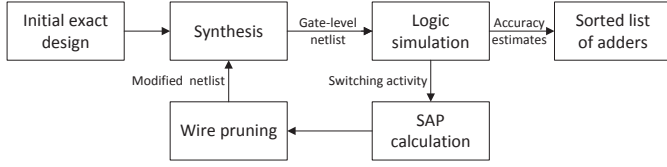


Fig. 1. CAD framework for Gate-Level Pruning.

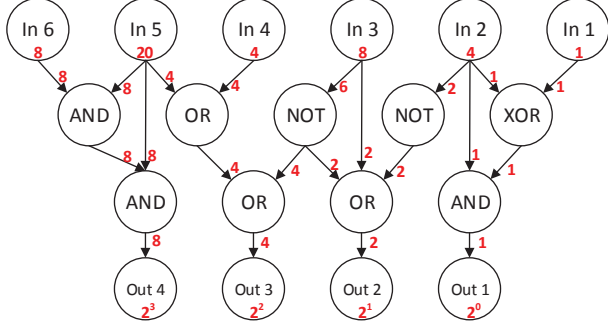


Fig. 2. Directed acyclic graph representation of a gate level netlist and the associated significance attribution.

such as gates, and whose edges are wires. The decision to prune a node is based on two criteria: the significance, which is a structural parameter, and the activity or toggle count. The nodes with the lowest Significance-Activity Product (SAP) are pruned first. By doing so, the error magnitude grows with the amount of pruning. Alternatively, depending on the application's requirement, the designer may choose to prune nodes according to the activity only, in order to minimize the error rate.

The activity of each wire is extracted from the SAIF file (Switching Activity Interchange Format) obtained through gate-level hardware simulations. This file contains the toggle count of each wire, as well as the time spent at the logic levels 0 and 1 respectively. In order to get an accurate activity estimation, the system should be simulated with an input stimulus representative of the *real operation* of the circuit. The more the simulation is realistic, the more the toggle count is accurate and leads to an efficient pruning.

The significance of each primary output is set by the designer depending on the application's requirement. In this paper, pruning is applied on several arithmetic circuits where each primary output is weighted by a power of two. It is therefore worth applying a weighted significance attribution, where each output bit position has a significance two times higher than the previous when moving from the LSB to the MSB. Reverse topological graph traversal is then performed to compute each nodes' significances as follows:

$$\sigma_i = \sum \sigma_{desc(i)} \quad (1)$$

where  $\sigma_i$  is the significance of the node  $i$  and  $\sigma_{desc(i)}$  is the significance of the direct descendants of node  $i$ . An example of weighted significance attribution is shown in Fig. 2.

Once the significance and activity is determined, the nodes, i.e. gates and their corresponding wires, are ranked according to their Significance Activity Product (SAP). The ones with the lowest SAP are disconnected from the verilog netlist, and a re-synthesis is performed in order to remove or replace the unconnected gates.

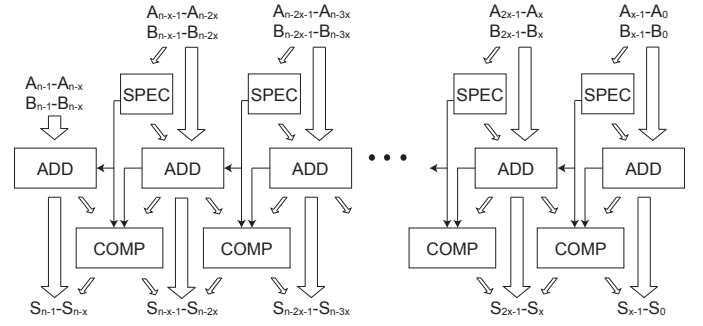


Fig. 3. General block diagram of an Inexact Speculative Adder (ISA). Each speculative segment consists of a carry speculator (SPEC), a regular adder (ADD) and an error compensation block (COMP).

## B. Inexact Speculative Adders

### 1) General Concept

The general block diagram of an Inexact Speculative Adder (ISA) adder is depicted in Fig. 3. An ISA splits the carry propagation chain in multiple paths executed concurrently. Each path consists of a carry speculator block (SPEC), a sub-adder block (ADD) and an error compensation block (COMP). For each of these SPEC-ADD-COMP paths, the different blocks have the following functions:

- **SPEC** – The speculator block generates a partial carry signal from a limited number of operand bits in a carry look-ahead approach and sourced by either a static or a dynamic input. When a propagate chain covers the full SPEC block, the exact carry cannot be speculated from the partial product and the output carry is guessed at the input value. As long propagate sequences are uncommon in uniform input distribution [4], the probability of fault decreases when increasing the size of this block.
- **ADD** – The sub-adder block calculates local sums from the speculated carry of the SPEC block.
- **COMP** – Without compensation, an internal overflow caused by an inconsistent carry could lead to a massive error. Therefore, the COMP block detects those speculation faults by comparing the carry generated from the SPEC with the carry-out coming from the prior ADD block. It then compensates faulty sums either by attempting to correct a few bits of the local sum or by reducing relative error over a few bits of the preceding sum.

The first speculative path, operating on the LSBs of the adder, does not have SPEC nor COMP blocks since it uses directly the adder carry-in. The achieved addition arithmetic is illustrated in Fig. 4.

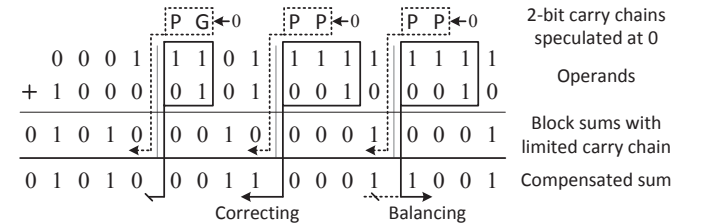


Fig. 4. Example of ISA addition arithmetic with 2-bit speculation, 1-bit correction and 1-bit error reduction. Faults only occur in the two right-hand paths. The 1<sup>st</sup> LSB of the central path can be corrected. The 1<sup>st</sup> LSB of the right path cannot be corrected, so the 1<sup>st</sup> MSB of the preceding sum is flipped.

### 2) Error Compensation

The COMP's error correction technique, introduced in [8], consists in incrementing or decrementing only a small group of LSBs of the local sum in order to compensate the erroneous speculated carry. In most cases, this can fully resolve carry errors. In the case where those stages are all in propagate modes, correction is impossible as it would lead to an internal overflow. In that case, the uncorrected bits, having a higher significance than the error bit, ensure a low relative error of the result. Using the COMP's error correction technique thus reduces both error rate and relative error. The correction hardware is executed concurrently to the local addition, thus this technique impacts minimally the critical path of the adder.

The COMP's error reduction technique consists in balancing a group of MSBs of the preceding sub-adders in opposite direction than the error. This technique, similarly as in [5], has been intensively employed in literature. But to avoid high relative errors and better control the worst-case error ( $RE_{MAX}$ ), it relies on large SPEC block directly lying in the critical path of the adder.

### 3) Design Strategy

The ISA offers a general topology of speculative compensated addition inclusive of the state-of-the-art and that allows an optimal balance between circuit and accuracy specifications.

A design methodology through a delay-accuracy approach is presented in Fig. 5. The adequate delay tradeoff is mainly obtained by sizing SPEC and ADD blocks, principal slack elements of the ISA. Then, the COMP's error correction and error reduction techniques enable to tune and fit the accuracy requirements at the cost of hardware overhead and with a minimum delay penalty for multiplexing the result on a few compensated bits.

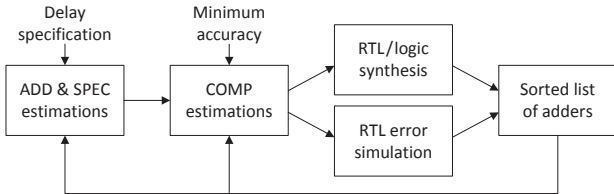


Fig. 5. CAD framework for ISA design.

Adders in literature describe particular cases of implementation excessively considering either performances or errors. In the ISA architecture, the speculation overhead can be traded for longer sub-adders while fitting the same delay requirement. It is then possible to use fewer speculative paths and limit the in-critical path speculation-compensation overhead to a few bits of each path while fitting the accuracy requirement. This approach allows notable improvements in circuit performances [8].

## IV. RESULTS AND COMPARATIVE STUDY

### A. Accuracy Metrics

In order to quantify the error produced by an inexact circuit, one has to choose one or more error metric depending on the application's requirement. The metrics used to characterize approximate adders are based on the relative error ( $RE$ ) of a sum, defined as:

$$RE = \left| \frac{S_{approx} - S_{correct}}{S_{correct}} \right| \quad (2)$$

where  $S_{approx}$  and  $S_{correct}$  are respectively the approximate and correct sums of an addition. In [1], three interesting metrics are defined:

- **Error Rate** – The error rate corresponds the ratio of erroneous computations over the entire set of computations and is defined as follow:

$$Error\ Rate = \frac{Number\ of\ erroneous\ computations}{Total\ number\ of\ computations} \quad (3)$$

- **Relative Error RMS ( $RE_{RMS}$ )** – The root mean square (RMS) of  $RE$  is a good estimator of accuracy and is interesting for many applications, particularly in image and video processing. It is defined as:

$$RE_{RMS} = \sqrt{\frac{1}{N} \sum_{n=1}^N RE_n^2} \quad (4)$$

- **Maximum Relative Error ( $RE_{MAX}$ )** –  $RE_{MAX}$  represents the largest relative error of an adder and defines its worst case accuracy. It is here obtained over a set of computations.

### B. Results

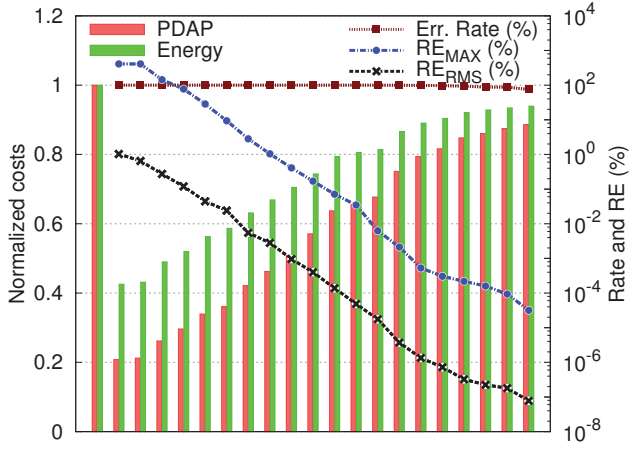
In order to perform a comparative study, both techniques have been applied to 32-bit adders synthesized in a 65 nm UMC technology library and all the designs have been simulated with a set of five million uniformly distributed random inputs. Fig. 6 show the error characteristics and the normalized costs in terms of energy and Power-Delay-Area Product (PDAP) for selected pruned and speculative adders respectively synthesized at 3.3 GHz. As timing constraint has been proved to strongly impact the gains with those techniques, the same work has been reproduced for adders running at 1.33 GHz in Fig. 7.

Only speculative adders with regular structures have been synthesized (i.e. 2x16, 4x8, 8x4 and 16x2 bit concurrent paths) with diverse error characteristics. For this reason, the displayed characteristics present steps corresponding changes of structure sizes. Generally, ISA adders built out of small sub-adders show high errors and high savings (on the left of the figures), whereas ISA with large sub-adders are preferred for low errors and lower savings (on the right of figures).

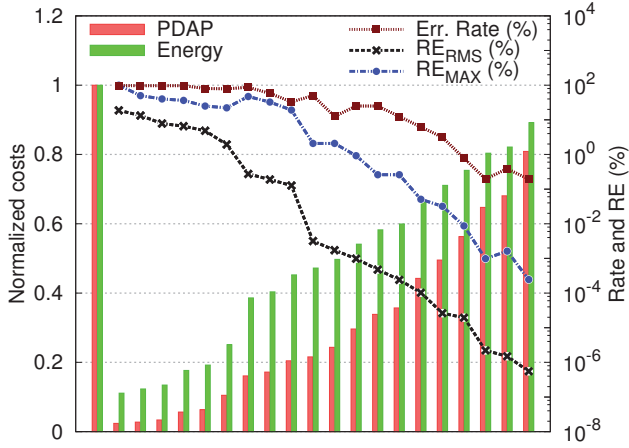
Fig. 6 and 7 clearly show that the two techniques have a different impact on the output quality. The error rate of pruned adders rapidly reaches 100 %, the reason being that in the first steps of the pruning process, some of the least significant outputs are removed. On the other hand, in speculative adders, a small speculation-correction overhead leads to a decrease of the error rate despite lower circuit efficiency.

For both techniques and frequencies, the  $RE_{RMS}$  and the  $RE_{MAX}$  grow with an exponentially trend versus circuit savings. The ISA adders on the left of figures have a low  $RE_{MAX}$ , but this one does not follow the same exponential trend as it is expensive to control. Thus, a gap appears between  $RE_{MAX}$  and  $RE_{RMS}$  when the constraints on the circuit become too high (at 1 % for 3.3 GHz and  $10^{-3}$  % at 1.6 GHz).

Timing constraint has a significant influence on the result obtained with the two techniques. Fig. 6 shows that at high frequency of 3.3 GHz, and for a relative PDAP cost of 0.42, the  $RE_{MAX}$  and the  $RE_{RMS}$  of the pruned adder are equal to 4 % and 0.008 % respectively. In comparison, the speculative adder having a similar PDAP has a  $RE_{MAX}$  of  $10^{-1}$  % and a  $RE_{RMS}$  of



(a) Pruned adders at 3.3 GHz



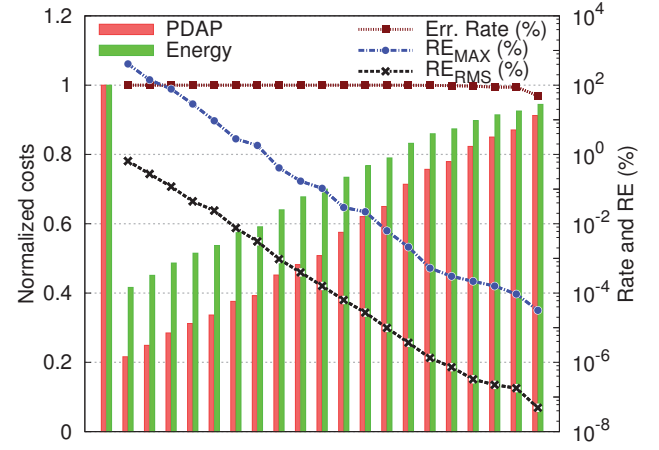
(b) Speculative adders at 3.3 GHz

Fig. 6. Error characteristics and normalized cost of 32-bit pruned (a) and speculative (b) adders synthesized at 3.3 GHz

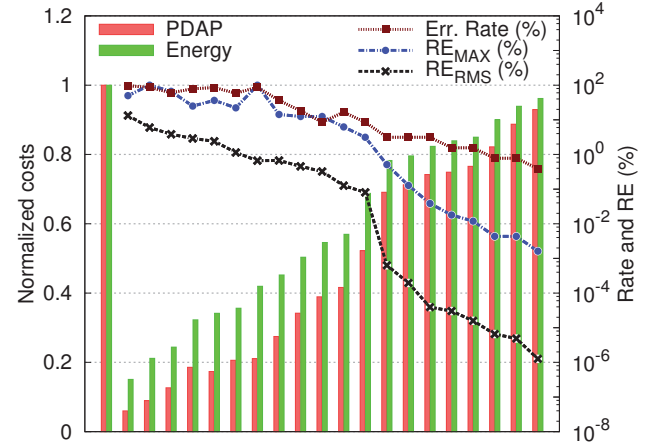
10<sup>-4</sup> %. This could lead to the conclusion that the speculation technique can achieve similar energy savings than the pruning technique, at a much higher accuracy level. However, Fig. 7 actually depicts the opposite trend when using a slightly lower frequency of 1.6 GHz. Hence, a more extensive comparative study might show that the two depicted design techniques might produce uncorrelated errors, and therefore could be combined to get additive savings.

## V. CONCLUSION

This paper reviewed and compared two well established techniques for generating approximate hardware: carry speculation and gate-level pruning. It has been shown that both can achieve up to 85 % PDAP reduction for a RMS relative error of 1 %. However the two techniques clearly have a different impact on the accuracy of the generated adders. Additionally, timing constraint significantly impacts the efficiency of such techniques: in the conducted experiments, speculative adder present fewer error than pruning for equivalent PDAP, at a 3.3 GHz frequency. On the other hand, the gate-level pruning is more efficient than carry speculation at 1.3 GHz. A more extensive study would certainly prove that the two techniques produce uncorrelated errors, and thus could be combined to further reduce power consumption, silicon area and critical path delay.



(a) Pruned adders at 1.6 GHz



(b) Speculative adders at 1.6 GHz

Fig. 7. Error characteristics and normalized cost of 32-bit pruned (a) and speculative (b) adders synthesized at 1.6 GHz

## REFERENCES

- [1] A. Lingamneni, C. Enz, J. L. Nagel, K. Palem, and C. Piguet, "Energy parsimonious circuit design through probabilistic pruning," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2011*, March 2011, pp. 1–6.
- [2] J. Schlachter, V. Camus, C. Enz, and K. Palem, "Automatic Generation of Inexact Digital Circuits by Gate-level Pruning," in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, May 2015.
- [3] T. Liu and S.-L. Lu, "Performance Improvement with Circuit-level Speculation," in *Microarchitecture, 2000. MICRO-33. Proceedings. 33rd Annual IEEE/ACM International Symposium on*, 2000, pp. 348–355.
- [4] N. Zhu, W.-L. Goh, and K.-S. Yeo, "An Enhanced Low-power High-speed Adder For Error-tolerant Application," in *Integrated Circuits (ISIC), Proc. of the 2009 12th International Symposium on*, Dec 2009, pp. 69–72.
- [5] M. Weber, M. Putic, H. Zhang, J. Lach, and J. Huang, "Balancing Adder for Error Tolerant Applications," in *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, May 2013, pp. 3038–3041.
- [6] N. Zhu, W.-L. Goh, G. Wang, and K.-S. Yeo, "Enhanced Low-power High-speed Adder for Error-tolerant Application," in *SoC Design Conference (ISOCC), 2010 International*, Nov 2010, pp. 323–327.
- [7] Y. Kim, Y. Zhang, and P. Li, "An Energy Efficient Approximate Adder with Carry Skip for Error Resilient Neuromorphic VLSI Systems," in *Computer-Aided Design (ICCAD), 2013 IEEE/ACM International Conference on*, Nov 2013, pp. 130–137.
- [8] V. Camus, J. Schlachter, and C. Enz, "Energy-Efficient Inexact Speculative Adder with High Performance and Accuracy Control," in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, May 2015.
- [9] J. Hu and W. Qian, "A New Approximate Adder with Low Relative Error and Correct Sign Calculation," in *Design, Automation and Test in Europe (DATE), 2015 IEEE Conference and Exhibition on*, March 2015.